Remarks

Claims 1-21 are pending.

Claim 21 has been added. See, for example, Claim 1, and Figures 2 and 3 and the corresponding disclosure.

A Fee Sheet and duplicate copy thereof accompany this Amendment.

OBJECTIONS TO THE SPECIFICATION

The Examiner objects to the title of the invention on the ground of it not being descriptive. The Examiner states that a new title is required that is clearly indicative of the invention to which the claims are directed. The Examiner, however, does not suggest a new title.

The title of the invention has been amended to recite "Computer System And Method For Secure Installation And Operation Of Software". It is submitted that the title, as amended, is as short and specific as possible, and is clearly indicative of the invention to which the claims are directed. Hence, it is submitted that the title is descriptive.

OBJECTIONS TO THE CLAIMS

The Examiner objects to Claims 1 and 11 on the ground of informalities. The Examiner states that the term "NT" "should incorporate with its full form."

Claim 1 recites "NT File Structure". It is submitted that this recital is clear to one of ordinary skill in the art. The specification (page 1, lines 17-18) defines "NT File Structure" as being equivalent to "NTFS". It is submitted that "NTFS" is also clear to one of ordinary skill in the art. It is further submitted that there is, thus, no requirement to change "NT" to "New Technology" as is apparently suggested by the Examiner, since Applicant's claimed invention is clear to one of ordinary skill in the art.

Claim 11 includes a similar recital as Claim 1. It is submitted that this recital is definite and clear for the same reasons.

Since the objections to Claims 1 and 11 have been dealt with, it is submitted that the objections to Claims 2-10 and 12-20, which depend from Claims 1 and 11, respectively, have also been dealt with.

REJECTIONS UNDER 35 U.S.C. § 112, ¶2

The Examiner rejects Claims 1-20 under 35 U.S.C. § 112, ¶2, on the ground of being indefinite.

The Examiner states that the term "installer" is unclear. The Examiner further states that the term installer is unclear "as ... used to install the file or the files are simply gets

copied where no installer is needed. In the claim language it seems that the files are only copying from one location to another."

Here, the Examiner appears to be of the position that "no installer is needed". It is respectfully submitted that this position is not well taken in view of the refined recitals (*emphasis added*) of Claim 1 of "employing an installer"; "writing a Primary Data Stream file to said NT File Structure logical volume *from said installer*"; and "writing said associated data to said NT File Structure logical volume as an Alternate Data Stream file *from said installer*". Hence, since the recited installer is employed, and since both a Primary Data Stream file and associated data as an Alternate Data Stream file are written to an NT File Structure logical volume from the installer, it is respectfully submitted that the position that "no installer is needed" cannot be supported.

Furthermore, the term "installer" is understood by those of ordinary skill in the art. See, for example, U.S. Patent No. 6,744,450 (<u>Zimniewicz et al.</u>) (col. 6, l. 45, reciting "existing installer technology") (of record).

Therefore, it is submitted that this recital of Claim 1 is definite and passes muster under Section 112, second paragraph.

As to the recital "associating data," the Examiner states that it is "unclear as to what data is being associated with Primary Data Stream". As was discussed above, Claim 1 recites "associating data with said Primary Data Stream file; and writing said associated data to said NT File Structure logical volume as an Alternate Data Stream file from said installer." Within the context of Claim 1, it is respectfully submitted that Applicant may recite "data" as broadly as possible without limiting said data. As is stated in Section 2173.04 of the Manual of Patent Examining Procedure (MPEP):

> Breadth of a claim is not to be equated with indefiniteness.
> *In re Miller*, 441 F.2d 689, 169 USPQ 597 (CCPA 1971).
> If the scope of the subject matter embraced by the claims is
> clear, and if applicants have not otherwise indicated that
> they intend the invention to be of a scope different from
> that defined in the claims, then the claims comply with 35
> U.S.C. 112, second paragraph.

MPEP § 2173.04. Hence, it is submitted that this recital of Claim 1 is definite and passes muster under Section 112, second paragraph.

As to Claim 11, the Examiner makes the same statements regarding the term "installer" as were discussed above in connection with Claim 1. Claim 11 recites that the installer cooperates with a processor to "write a Primary Data Stream file to said NT File

Structure logical volume, associate data with said Primary Data Stream file, and write said associated data to said NT File Structure logical volume as an Alternate Data Stream file." For similar reasons as were discussed above in connection with Claim 1, it is respectfully submitted that the position that "no installer is needed" cannot be supported. Furthermore, the term "installer" is understood by those of ordinary skill in the art. See, for example, U.S. Patent No. 6,744,450 (Zimniewicz et al.) (col. 6, l. 45, reciting "existing installer technology") (of record). Accordingly, it is submitted that this recital of Claim 11 is definite and passes muster under Section 112, second paragraph.

Since the rejections to Claims 1 and 11 have been dealt with, it is submitted that the rejections of Claims 2-10 and 12-20, which depend from Claims 1 and 11, respectively, have also been dealt with.

REJECTIONS UNDER 35 U.S.C. § 101

The Examiner rejects Claims 1-10 under 35 U.S.C. § 101 on the ground that the claimed invention is directed to non-statutory subject matter.

The Examiner states that Claim 1 recites a "method for secure installation and operation of software, representing functional descriptive matter without a computer readable medium or computer implemented method, method per se is not tangibly embodied." This statement and rejection are respectfully traversed.

Claim 1 is an independent method claim which recites, *inter alia*, a method for secure installation and operation of software comprising: employing an NT File Structure logical volume; employing an installer; writing a Primary Data Stream file to the NT File Structure logical volume from the installer; associating data with the Primary Data Stream file; and writing the associated data to the NT File Structure logical volume as an Alternate Data Stream file from the installer.

The Examiner's statement that Claim 1 is "without a computer readable medium or computer implemented method" is misplaced.

> When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized.

MPEP § 2106. *See In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (claim to data structure stored on a computer readable medium that increases computer efficiency held statutory); and *In re Warmerdam*, 33 F.3d 1354, 1360-61, 31 USPQ2d 1754, 1759 (Fed. Cir. 1994) (claim to computer having a specific data structure stored in memory

held statutory product-by-process claim). Claim 1 recites writing a Primary Data Stream file to an NT File Structure logical volume from an installer, and writing associated data to such NT File Structure logical volume as an Alternate Data Stream file from such installer. Hence, a Primary Data Stream file and associated data as an Alternate Data Stream file are recorded on a memory, such as the recited NT File Structure logical volume. Therefore, it is submitted that Claim 1 recites statutory subject matter.

Since the rejection to Claim 1 has been dealt with, it is submitted that the rejections of Claims 2-10, which depend from Claim 1, have also been dealt with.

REJECTIONS UNDER 35 U.S.C. § 102(a)

The Examiner rejects Claims 1, 3-5, 8-12, 14-16, 19 and 20 on the ground of being anticipated by "Phishing in Alternate Data Streams" (Berghel et al.).

Berghel et al. discloses Primary and alternate data streams (ADSs) in the New Technology File System (NTFS) of Microsoft. This reference also discloses (page 7) that there is "malware" that takes advantage of ADSs (*e.g.*, W2k.stream).

Claim 1 recites, *inter alia*, a method for secure installation and operation of software comprising: employing an NT File Structure logical volume; employing an installer; writing a Primary Data Stream file to the NT File Structure logical volume from the installer; associating data with the Primary Data Stream file; and writing the associated data to the NT File Structure logical volume as an Alternate Data Stream file from the installer.

Claim 1 recites employing an *installer*; writing a Primary Data Stream file to an NT File Structure logical volume from such *installer*; *and* writing associated data with the Primary Data Stream file and to the NT File Structure logical volume as an Alternate Data Stream file from such *installer*.

Berghel et al. teaches and suggests a W2K.Stream <u>virus</u>. When this <u>virus</u> infects a file it replaces a host application with <u>itself</u>. Basically, the virus implements the simplest possible virus infection by overwriting the host program with <u>its own code</u>. A <u>virus</u> is completely different from the recited *installer* and does not install or upgrade files in a traditional sense. Instead, a virus usually overwrites existing files or exists as a parasite within existing files. This view is confirmed by the express teachings of Berghel et al., which states (page 6, at <u>www.sarc.com/avcenter/venc/data/w2k.stream.html</u> (of record as Cite No. B, "W2K.Stream")):

> The virus is basically a new subclass of companion viruses, a "stream companion" virus. When the virus infects a file it replaces the host application with itself. Basically the virus implements the

> simplest possible virus infection by overwriting the host program
> with its own code.

Hence, a <u>virus</u> replaces a host application with <u>itself</u>. This view is also supported by Exhibit 4 of the Petition To Make Special Pursuant to 37 CFR 1.102(d), filed on October 21, 2004 (of record), which provides a definition of "virus" namely a "program that can 'infect' other programs by modifying them to include a, possibly evolved, copy of itself. A program that infects a computer by att[]aching itself to another program, and propagating itself when that program is executed."

Furthermore, a malware <u>virus</u> is a "program or piece of code that is loaded onto your computer without your knowledge and runs against your wishes." *See* Exhibit 5 of the Petition To Make Special Pursuant to 37 CFR 1.102(d) (of record). This is completely different from a method for *secure* installation and operation of software.

The reference does not teach or suggest the refined recital of employing an *installer*; writing a Primary Data Stream file to an NT File Structure logical volume from such *installer*; *and* writing associated data with the Primary Data Stream file and to the NT File Structure logical volume as an Alternate Data Stream file from such *installer*.

The Examiner admits that <u>Berghel et al.</u> is silent on the recited installer. The Examiner states that this feature is "deemed to be inherent to Berghel system." The Examiner further states that <u>Berghel et al.</u> (page 4) discloses "ADS contains binary executables" and that it would be "inoperative if the installer is not present to provide ADS from PDS that includes [an] executable file." These statements are respectfully traversed.

It is submitted that the Examiner improperly uses impermissible hindsight to reach the above conclusions in view of Applicant's refined recital in Claim 1. Furthermore, <u>Berghel et al.</u> (page 4) makes clear that:

> We now rename <calc.exe> as the ADS, <d.exe>, and associate it
> with the empty text file <test.txt>
>
> C:\...\test>type c:\windows\system32\calc.exe > .\test.txt:d.exe
>
> and execute the ADS directly
>
> C:\...\test>start .\test.txt:d.exe

This also has nothing to do with any installer within the context of Claim 1 as is understood by those of ordinary skill in the art. Here, the user is merely renaming files through a

command line of a DOS command prompt window. This also does not teach or suggest an installer within the context of Claim 1.

Accordingly, for the above reasons, Claim 1 patentably distinguishes over the reference.

Claims 3-5 and 8-10 depend either directly or indirectly from Claim 1 and patentably distinguish over the reference for the same reasons.

Furthermore, Claim 5 recites creating a Primary Data Stream directory chain; writing the Primary Data Stream directory chain to the NT File Structure logical volume from the *installer*; writing the Primary Data Stream file to the Primary Data Stream directory chain in the NT File Structure logical volume from the *installer*; associating the data with the Primary Data Stream directory chain or the Primary Data Stream file by creating and closing the Alternate Data Stream file; and *installing* the associated data to the NT File Structure logical volume as the Alternate Data Stream file from the *installer*.

Since the reference neither teaches nor suggests the refined recital of Claim 1, it clearly neither teaches nor suggests these additional limitations which further patentably distinguish over the reference.

Again, the Examiner's reliance on the command line of a DOS command prompt window (Berghel et al. (page 2)) has nothing to do with any installer within the context of the claims and as understood by those of ordinary skill in the art.

Furthermore, Claim 8 recites employing as the associated data first data; employing as the Alternate Data Stream file a first Alternate Data Stream file; employing second data; associating the second data with the Primary Data Stream file; and writing the associated second data to the NT File Structure logical volume as a second Alternate Data Stream file from the *installer*.

Since the reference neither teaches nor suggests the refined recital of Claim 1, it clearly neither teaches nor suggests these additional limitations which further patentably distinguish over the reference.

Again, the Examiner's apparent reliance on the command line of a DOS command prompt window, as was discussed above in connection with Claim 1, has nothing to do with any installer within the context of the claims and as understood by those of ordinary skill in the art.

Furthermore, Claim 10 recites employing an *installation* file; defining in the installation file a Primary Data Stream directory chain, the Primary Data Stream file, the Alternate Data Stream file, and at least one information file; displaying the at least one

information file from the installation file; creating the Primary Data Stream directory chain in the NT File Structure logical volume; copying the Primary Data Stream file from the installation file to the Primary Data Stream directory chain in the NT File Structure logical volume; and copying the Alternate Data Stream file from the installation file to the Primary Data Stream directory chain in the NT File Structure logical volume.

Since the reference neither teaches nor suggests the refined recital of Claim 1, its clearly neither teaches nor suggests these additional limitations which further distinguish over the reference.

Again, the Examiner's reliance on the command line of a DOS command prompt window (Berghel et al. (pages 2, 3 and 9)) has nothing to do with any installer within the context of the claims and as understood by those of ordinary skill in the art.

Claim 11 is an independent claim which recites, *inter alia*, a computer system for secure installation and operation of software comprising: a processor; a first drive adapted for access by the processor; a second drive adapted for access by the processor, the second drive including an NT File Structure logical volume; and an installer operatively associated with the first drive, the installer cooperating with the processor to write a Primary Data Stream file to the NT File Structure logical volume, associate data with the Primary Data Stream file, and write the associated data to the NT File Structure logical volume as an Alternate Data Stream file.

For reasons that were discussed above in connection with Claim 1, a virus is completely different from the recited *installer* and does not install or upgrade files in a traditional sense. Instead, a virus usually overwrites existing files or exists as a parasite within existing files, and replaces a host application with itself. Furthermore, a malware virus is a "program or piece of code that is loaded onto your computer without your knowledge and runs against your wishes." This is completely different from a computer system for *secure* installation and operation of software.

The Examiner admits that Berghel et al. is silent on the recited installer. The Examiner states that this feature is "deemed to be inherent to Berghel system." The Examiner further states that Berghel et al. (page 4) discloses "ADS contains binary executables" and that it would be "inoperative if the installer is not present to provide ADS from PDS that includes [an] executable file." These statements were respectfully traversed, above, in connection with Claim 1. Again, these statements have nothing to do with any installer within the context of the claims and as understood by those of ordinary skill in the art. The user in Berghel et al. is merely renaming files through a command line of a DOS

- 13 -

command prompt window. This also does not teach or suggest an installer within the context of Claim 11.

The reference does not teach or suggest the refined recital of a computer system for *secure* installation and operation of software comprising: an *installer* operatively associated with a first drive, such *installer* cooperating with a processor to write a Primary Data Stream file to an NT File Structure logical volume, associate data with such Primary Data Stream file, and write such associated data to such NT File Structure logical volume as an Alternate Data Stream file.

Therefore, for the above reasons, Claim 11 patentably distinguishes over the reference.

Claims 12, 14-16, 19 and 20 depend either directly or indirectly from Claim 11 and patentably distinguish over the reference for the same reasons.

Furthermore, Claim 12 recites that the NT File Structure logical volume includes a directory chain or a system directory; and that the *installer* installs the Primary Data Stream file in the directory chain or the system directory of the NT File Structure logical volume.

Since the reference neither teaches nor suggests the refined recital of Claim 11, it clearly neither teaches nor suggests these additional limitations which further patentably distinguish over the reference.

Again, the Examiner's reliance on the command line of a DOS command prompt window (Berghel et al. (pages 2, 3 and 9)) has nothing to do with any installer within the context of the claims and as understood by those of ordinary skill in the art.

Furthermore, Claim 16 recites that the *installer* cooperates with the processor to create a Primary Data Stream directory chain, to write the Primary Data Stream directory chain to the NT File Structure logical volume, to write the Primary Data Stream file to the Primary Data Stream directory chain in the NT File Structure logical volume, to associate the data with the Primary Data Stream directory chain or the Primary Data Stream file, and to install the associated data to the NT File Structure logical volume as the Alternate Data Stream file.

Since the reference neither teaches nor suggests the refined recital of Claim 11, it clearly neither teaches nor suggests these additional limitations which further patentably distinguish over the reference.

Again, the Examiner's reliance on the command line of a DOS command prompt window (Berghel et al. (pages 2, 3 and 9)) has nothing to do with any installer within the context of the claims and as understood by those of ordinary skill in the art.

Furthermore, Claim 20 recites that the processor includes a display; that the *installer* comprises an *installation* file including a Primary Data Stream directory chain, the Primary Data Stream file, the Alternate Data Stream file, and at least one information file; and that the *installer* cooperates with the processor to display the at least one information file from the installation file to the display, to create the Primary Data Stream directory chain in the NT File Structure logical volume, to copy the Primary Data Stream file from the installation file to the Primary Data Stream directory chain in the NT File Structure logical volume, and to copy the Alternate Data Stream file from the installation file to the Primary Data Stream directory chain in the NT File Structure logical volume.

Claim 20 further patentably distinguishes over the reference for similar reasons as were discussed above in connection with Claim 10.

Newly added Claim 21 is an independent method claim which recites, *inter alia*, a method for secure installation and operation of software comprising: employing a computer-readable medium including an NT File Structure logical volume; employing an installer; writing a Primary Data Stream file to the NT File Structure logical volume of the computer-readable medium from the installer; associating data with the Primary Data Stream file; and writing the associated data to the NT File Structure logical volume of the computer-readable medium as an Alternate Data Stream file from the installer.

For similar reasons as were discussed above in connection with Claim 1, the reference does not teach or suggest employing an *installer*; writing a Primary Data Stream file to an NT File Structure logical volume of a computer-readable medium from such *installer*; associating data with such Primary Data Stream file; and writing such associated data to such NT File Structure logical volume of such computer-readable medium as an Alternate Data Stream file from such *installer*.

Therefore, for the above reasons, Claim 21 patentably distinguishes over the reference.

REJECTIONS UNDER 35 U.S.C. § 103(a)

The Examiner rejects Claims 2, 6, 7, 13, 17 and 18 on the ground of being unpatentable over Berghel et al. in view of U.S. Patent No. 6,744,450 (Zimniewicz et al.).

Zimniewicz et al. discloses a system and method for a suite integration toolkit (SIT) allowing for the provision and display of a set of installation actions.

Zimniewicz et al., which does not disclose any Primary Data Stream file or any Alternate Data Stream file, adds nothing to Berghel et al. regarding writing a Primary Data Stream file or an Alternate Data Stream file to an NT File Structure logical volume from an installer to render Claims 1 or 11 unpatentable.

Furthermore, Claim 6 recites employing an *installation* file comprising the Primary Data Stream file, the Alternate Data Stream file, installation instructions, the Primary Data Stream directory chain, and an End User License Agreement.

Claim 6 depends directly from Claim 5 and indirectly from Claim 1 and includes all of the limitations of those claims. Since the references neither teach nor suggest the refined recital of Claim 5, they clearly neither teach nor suggest these additional limitations which further patentably distinguish over the references.

Again, the Examiner's reliance on the command line of a DOS command prompt window (Berghel et al. (page 2)) has nothing to do with any installer within the context of the claims and as understood by those of ordinary skill in the art.

Furthermore, Claim 17 recites that the *installer* comprises an *installation* file comprising the Primary Data Stream file, the Alternate Data Stream file, installation instructions, a Primary Data Stream directory chain, and an End User License Agreement.

Claim 17 further patentably distinguishes over the references for similar reasons as were discussed above in connection with Claim 6.

Furthermore, Claim 18 recites that the processor includes a display; and that the *installer* cooperates with the processor to display the installation instructions and the End User License Agreement on the display.

Since the references neither teach nor suggest the refined recital of Claims 11 and 17, they clearly neither teach nor suggest these additional limitations which further patentably distinguish over the references.

Again, the Examiner's reliance on the command line of a DOS command prompt window (Berghel et al. (pages 2, 3 and 9)) has nothing to do with any installer within the context of the claims and as understood by those of ordinary skill in the art.

The Examiner states that the "prior art made of record and not relied upon is considered pertinent to applicant's disclosure." Here, the Examiner does not identify any particular reference(s) of the references of record.

For reasons discussed herein, Claims 1-21 patentably distinguish over the references of record.

Reconsideration and early allowance are respectfully requested.

Respectfully submitted,

Kirk D. Houser
Registration No. 37,357
Attorney for Applicant

(412) 566-6083